

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian terkait pengenalan wajah yang sebelumnya telah dikerjakan oleh banyak peneliti dengan berbagai metode yang sudah dirancang. Hasil penelitian tersebut kemudian dipublikasikan agar dapat membantu penelitian selanjutnya. Berikut pustaka hasil penelitian terdahulu mengenai pengenalan wajah yang penulis tinjau untuk menilai kebaruan penelitian yang dilakukan.

Patrik, dkk. melakukan penelitian tentang metode baru untuk mengenali wajah [12]. Penelitian tersebut membandingkan beberapa metode pengenalan wajah dengan CNN. Metode yang dibandingkan adalah Eigenfaces dengan Principle Component Analysis (PCA) [13], Local Binary Pattern Histogram (LBPH) [14], dan k-Nearest Neighbor (kNN) [15]. Dalam penerapannya, peneliti tersebut menggunakan 7 *layer* yang dibuat sendiri seperti pada Gambar 2.1. Bahasa yang digunakan dalam melakukan penelitian tersebut adalah C++ dan Python serta *tools* yang digunakan adalah Matlab. Gambar yang digunakan untuk pengujian berasal dari *The ORL Database of Faces* yang memiliki 40 kategori dengan 10 gambar yang berbeda-beda untuk setiap kategorinya. Untuk menentukan kemiripan antara data uji dengan data *training*, peneliti tersebut menggunakan Eucliden *distance* namun tidak disebutkan berapa batas bawah yang digunakan sebagai patokan tingkat kemiripan dalam melakukan *testing*.

Layer	Type	Properties
Layer 1	Input	32×32
Layer 2	1 Convolutional	16 feature maps 3×3 kernel dimension
Layer 3	Pooling	2×2 kernel dimension probability 0.25
Layer 4	2 Convolutional	16 feature maps 3×3 kernel dimension
Layer 5	Pooling	2×2 kernel dimension probability 0.25
Layer 6	Fully-connected	3000 neurons
Layer 7	Fully-connected (Softmax)	40 neurons (classes)

Gambar 2.1 CNN Layer[12]

Berdasarkan hasil penelitiannya, peneliti tersebut memberikan kesimpulan bahwa CNN memiliki akurasi yang lebih baik dibandingkan dengan 3 metode lainnya. Adapun kekurangan dari metode CNN ini yaitu dibutuhkan kumpulan data latih yang banyak untuk memperoleh hasil yang lebih akurat. Semakin banyak data, maka semakin besar beban komputasi dan penggunaan memori yang semakin meningkat.

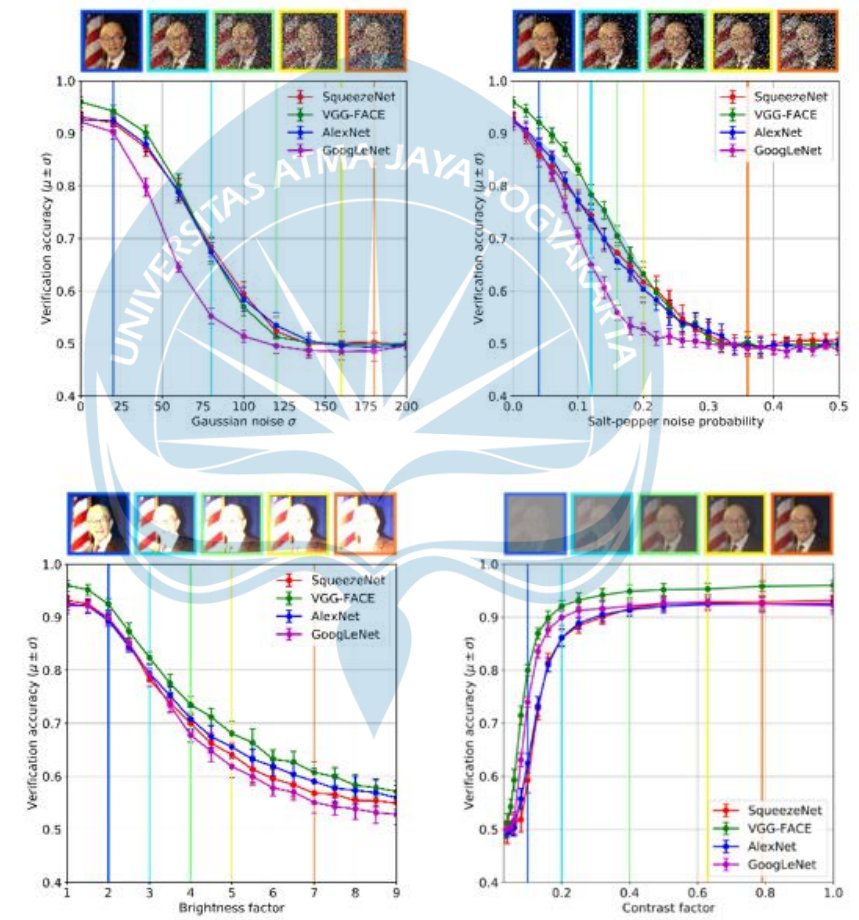
Klemen, dkk. meneliti tentang kelebihan dan kekurangan *Deep Learning* dalam mengenali wajah terhadap degradasi gambar [16]. Pada penelitian tersebut, terdapat beberapa arsitektur model yang digunakan yaitu AlexNet [17], VGGFace [18], Google Net [19], dan Squeeze Net [20]. VGGFace menggunakan arsitektur VGG 16 dan Google Net menggunakan Inception Versi 3 (V3). Pengujian dilakukan dengan menggunakan model evaluasi Cosine Distance dan dataset VGGFace. Dataset tersebut memiliki sekitar  $2.6 \times 10^6$  gambar dengan 2622 kategori atau sekitar 1000 gambar per kategorinya. Namun tidak semua gambar digunakan, hanya  $1.8 \times 10^6$  gambar yang digunakan dalam penelitian tersebut.

Pada pengujiannya, peneliti tersebut melakukan perubahan terhadap kualitas gambar. Perubahan yang dilakukan adalah *Blur*, *Compression*, *Gaussian Noise*, *Salt-and-pepper noise*, *Brightness*, *Contrast*, dan *Missing Data*, ilustrasi dapat dilihat pada Gambar 2.2. Berdasarkan hasil pengujian yang telah dilakukan, tidak ada arsitektur yang mampu untuk mengenali seluruh kondisi perubahan kualitas tersebut. VGGFace mampu mengenali *noise* gambar namun lemah saat diuji dengan penurunan cahaya gambar, sedangkan GoogleNet berjalan sebaliknya.



Gambar 2.2 Ilustrasi Perubahan Kualitas Gambar [16]

Penurunan kualitas gambar yang dilakukan oleh peneliti tersebut dibagi dalam 5 tahap. Tahapan penurunan diawali dengan gambar normal sampai kualitas terendah yang dapat diterapkan pada gambar tersebut. Hasil dari pengujian tersebut dapat dilihat pada Gambar 2.3. Berdasarkan gambar tersebut penulis menyimpulkan bahwa pada pengujian tahap 1 dan 2, VGGFace memiliki akurasi paling tinggi dibandingkan dengan arsitektur lainnya, namun semakin melemah pada saat memasuki tahap yang ke 3.



*Gambar 2.3 Hasil Pengujian Penurunan Kualitas Gambar[16]*

Suleman, dkk. melakukan penelitian mengenai pengenalan wajah menggunakan CNN dan merancang penggunaan model tersebut ke dalam sebuah kaca mata pintar [21]. Penelitian tersebut menggunakan bahasa Python dan Alexnet sebagai arsitektur utamanya. Data yang digunakan untuk mengenai satu kategori atau satu orang adalah 2.500 wajah dengan ekspresi dan pose yang berbeda-beda. Data tersebut dapat dilihat pada Gambar 2.4.

Sebelum gambar diproses, peneliti tersebut menggunakan algoritma AdaBoost Haar-like Features [22] untuk memotong bagian wajah. Algoritma tersebut hanya dapat mengenali wajah pada gambar dengan warna *grayscale* atau hitam putih. Setelah wajah ditemukan, gambar dikembalikan lagi menjadi RGB dan *resize* menjadi 227 x 227 *piksel*. Hasil akurasi yang diperoleh mencapai 98.5% dengan melakukan 1.690 iterasi.



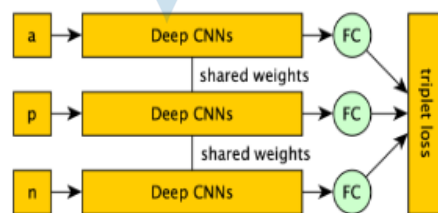
Gambar 2.4 Data Gambar Untuk Satu Kategori[21]

Selain pembuatan model *Deep Learning*, peneliti tersebut juga memberikan rancangan mengenai *hardware* yang dibutuhkan untuk melakukan pengenalan wajah. Beberapa komponennya adalah kacamata dengan kamera kecil, sebuah memori untuk menyimpan model dan *dataset*, Raspberry Pi sebagai unit prosesnya, dan O-Led Display untuk menampilkan hasilnya. Adapun proses pengenalan wajah yang dilakukan oleh kacamata pintar tersebut yaitu menerima *inputan* berupa gambar atau video, melakukan deteksi wajah dengan algoritma AdaBoost dan memotong bagian wajah, memasukkan gambar tersebut ke dalam model yang telah dibuat dan diproses dengan unit proses sehingga menghasilkan nilai kemiripan, dan menampilkan hasil dari pengenalan wajah tersebut berdasarkan keluaran dari model.

Katarina dan Hairul melakukan penelitian mengenai verifikasi wajah dengan arsitektur jaringan Siamese [23]. Penelitian yang dilakukan menggunakan model ResNet 50 yang sudah dimodifikasi dan bahasa yang digunakan adalah Python. Perubahan ResNet 50 [24] yang dilakukan adalah

pembuangan *average pooling layer* dan menambah operasi L2-Normalize dan perkalian skala *alpha* pada *fully connected layer*. Dalam pembuatan modelnya, penelitian tersebut menggunakan VGGFace2 [25] sebagai *dataset training*. Hasil akhir dari model yang dibuat berupa *feature* vektor atau biasa disebut sebagai *embedding*.

Untuk menentukan tingkat akurasi dari model yang dibuat, peneliti tersebut menggunakan jaringan Siamese. Jaringan tersebut memasukkan 2 sampai 3 *inputan* ke dalam model ResNet 50, kemudian dilakukan Triplet Loss atau melakukan perbandingan antar *embedding* yang dihasilkan. Jaringan Siamese dapat dilihat pada Gambar 2.5. Pada penelitian tersebut, *inputan* yang dimasukkan berupa gambar *testing*, gambar *positive*, dan gambar *negative*. Gambar *testing* merupakan gambar yang akan diujikan, gambar *positive* merupakan gambar yang mirip dengan gambar *testing*, sedangkan gambar *negative* merupakan gambar yang berbeda dengan gambar *testing*. Nilai *embedding* dari gambar *testing* akan dibandingkan dengan nilai *embedding* dari gambar *positive*, kemudian dibandingkan lagi dengan gambar *negative*. Perbandingan yang dilakukan menggunakan persamaan Euclidean *distance*. Apabila nilai *negative* lebih besar dari *positive* ditambah margin (nilai yang ditentukan sendiri) maka nilai tidak digunakan, apabila sebaliknya maka *weight* model akan diperbaharui.



Gambar 2.5 Arsitektur Jaringan Siamese [23]

Dari beberapa penelitian terdahulu yang sudah penulis rangkum, terdapat beberapa aspek yang dapat diambil sebagai pembandingan. Beberapa aspek tersebut yaitu algoritma pengenalan wajah, model jaringan, gambar yang digunakan untuk *training* per kategorinya, metode evaluasi, tahap *preprocessing*, dan implementasi terhadap model yang telah dibuat. Nilai-nilai pembandingan tersebut penulis rangkum di dalam Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian

Unsur Pembeding	(Patrik, dkk.) [12]	(Klemen, dkk.) [16]	(Suleman, dkk.) [21]	(Kartarina, Hairu) [23]	Rio*
Algoritma Pengenalan Wajah	CNN, EigenFaces with PCA, LBPH, dan KNN	CNN	CNN	CNN	CNN
Model Jaringan	-	AlexNet, VGG 16 VGGFace, Inception V3 Google Net, Squeeze Net	AlexNet	Residual Network (ResNet) 50	Squeeze-and-Excitation Networks (SeNet) 50
Pre-train Model (weight)	-	VGGFace	-	VGG-Face2	VGG-Face2
Bahasa Pemrograman	C++ dan Python	-	Python	Python	Python
Metode Evaluasi	Euclidean	Cosine	-	Euclidean	Cosine
Pre-processing	-	-	AdaBoost Haar-Like Features	-	ResNet 10 Caffe Model

\*Penelitian pada laporan ini